(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2001－84235
（P2001－84235A）
(43)公開日　平成13年３月30日(2001.3.30)

(51)Int.Cl.⁷	識別記号	Ｆ Ｉ		テーマコード（参考）
Ｇ０６Ｆ　15/177	６８２	Ｇ０６Ｆ　15/177	６８２Ｆ	５Ｂ０４５
12/00	５７２	12/00	５７２Ａ	５Ｂ０６０

審査請求　有　　請求項の数７　ＯＬ　（全６頁）

(21)出願番号　特願平11－256768

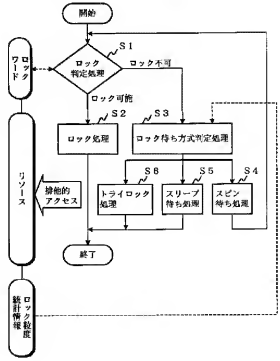
(22)出願日　平成11年９月10日(1999.9.10)

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(72)発明者　田辺　純
東京都港区芝五丁目７番１号　日本電気株式会社内
(74)代理人　100088959
弁理士　境　廣巳

Ｆターム(参考)　5B045 EE03 EE19
5B060 CD17 KA02 KA04

(54)【発明の名称】　ロック粒度統計情報を利用した排他制御方法及びプログラムを記録した機械読み取り可能な記録媒体

(57)【要約】
【課題】　共有メモリ方式の多重プロセッサ計算機システムにおいて、ロック待ち発生時に発生するオーバヘッド時間を低減させることにより、システム性能を向上させる。
【解決手段】　ロック要求されたリソースＲをロックできない場合、リソースＲのロック粒度統計情報（ロック時間の平均値等を含む）およびロック要求元がリソースＲを使用しない他に行うべき処理を有しているか否かを示すフラグ情報を利用してロック待ち方式をスピン方式，スリープ方式，トライロック方式の何れにするかを決定する。ロック待ち方式をスピン方式にした場合は、更に、ロック粒度統計情報に基づいてスピン待ち時間を決定する。



1

【特許請求の範囲】

【請求項１】　複数のプロセッサがメインメモリを共有する共有メモリ方式の多重プロセッサ計算機システムにおいて、

前記複数のプロセッサによって共有されるリソースがロックされる毎にロック粒度を算出すると共に、該算出したロック粒度に基づいて前記リソースのロック粒度統計情報を更新し、

前記リソースに対するロック待ちが発生したとき、前記リソースのロック粒度統計情報に基づいてロック待ち方式を決定することを特徴とするロック粒度統計情報を利用した排他制御方法。

【請求項２】　請求項１記載のロック粒度統計情報を利用した排他制御方法において、

前記ロック粒度統計情報は、ロック粒度の平均値を含むことを特徴とするロック粒度統計情報を利用した排他制御方法。

【請求項３】　請求項１記載のロック粒度統計情報を利用した排他制御方法において、

前記ロック粒度統計情報に基づいて決定するロック待ち方式は、スピン待ち時間経過後に再度前記リソースに対するロックを試みるスピン方式或いはロック要求元をスリープさせるスリープ方式であることを特徴とするロック粒度統計情報を利用した排他制御方法。

【請求項４】　請求項３記載のロック粒度統計情報を利用した排他制御方法において、

ロック待ち方式をスピン方式と決定した場合は、前記リソースのロック粒度統計情報に基づいて前記スピン待ち時間を決定することを特徴とするロック粒度統計情報を利用した排他制御方法。

【請求項５】　複数のプロセッサがメインメモリを共有する共有メモリ方式の多重プロセッサ計算機システムにおいて、

前記複数のプロセッサによって共有されるリソースがロックされる毎にロック粒度を算出すると共に、該算出したロック粒度に基づいて前記リソースのロック粒度統計情報を更新し、

前記リソースに対するロック待ちが発生したとき、前記リソースのロック粒度統計情報と、ロック要求元が前記リソースを使用しない処理を行えるか否かを示す情報とに基づいて前記ロック待ち方式を決定し、該決定したロック待ち方式がスピン待ち時間経過後に再度前記リソースに対するロックを試みるロック待ち方式である場合は前記リソースのロック粒度統計情報に基づいて前記スピン待ち時間を決定することを特徴とするロック粒度統計情報を利用した排他制御方法。

【請求項６】　請求項５記載のロック粒度統計情報を利用した排他制御方法において、

前記ロック粒度統計情報に基づいて決定するロック待ち方式は、スピン待ち時間経過後に再度前記リソースに対するロックを試みるスピン方式，ロック要求元をスリープさせるスリープ方式或いは要求元に他の処理を行わせるトライロック方式であることを特徴とするロック粒度統計情報を利用した排他制御方法。

2

【請求項７】　コンピュータに、

リソースがロックされる毎にロック粒度を算出すると共に、該算出したロック粒度に基づいて前記リソースのロック粒度統計情報を更新する処理と、

前記リソースに対するロック待ちが発生したとき、前記リソースのロック粒度統計情報に基づいてロック待ち方式を決定する処理とを行わせるためのプログラムを記録した機械読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、共有メモリ方式の多重プロセッサ計算機システムにおける排他制御方法に関し、特に、排他制御のオーバヘッド時間による計算機システムの性能低下を低減させることができるロック粒度統計情報を利用した排他制御方法に関する。

【０００２】

【従来の技術】複数のプロセッサがメインメモリを共有する共有メモリ方式の多重プロセッサ計算機システムにおいては、一般に、プロセス（手続き）によるリソース（プロセスが参照や変更などを行う資源や、その資源に対する一連の処理のかたまり）の逐次的（排他的）な使用を保証するため、ロックワードが格納されたメモリ上の領域に対する参照と設定とを１マシンクロックで行うテストアンドセット命令を利用している。ロックワードは、一般的にはメインメモリ上に置かれることが多いが、計算機システムによってはメインメモリではなく特定（専用）のメモリ上に置かれる場合もある。

【０００３】テストアンドセット命令を利用した従来の一般的な排他制御方法は、次のようなものであった。例えば、或るプロセスＰａが或るリソースＲａの排他的な使用を要求すると、オペレーティングシステムがテストアンドセット命令により、リソースＲａに対応するロックワードを参照してリソースＲａがロック状態であるか否かを調べると共にリソースＲａに対応するロックワードをロック状態を示す値に変更する。

【０００４】そして、ロックワードがロック状態でないことを示している場合は、プロセスＰａにロック成功を通知し、リソースＲａの排他的な使用を許可する。これに対して、ロックワードがロック状態であることを示している場合には、オペレーティングシステムは、リソースをロックできなかった場合に行う処理（ロック待ち発生時の処理）を行う。ロック待ち発生時に行う処理の方式（ロック待ち方式）には、（１）ロックが取得できるまでスピンしながら待ち続けるスピン方式、（２）他に行うべき処理がないのでプロセスＰａをスリープさせ、他のプロセスのためにプロセッサ（ＣＰＵ）を開放する

スリープ方式、（３）リソースＲａを使用しない他に行うべき処理があるので、リソースＲａに対するロックをあきらめ、プロセスＰａにリソースＲａを使用しない他の処理を行わせるトライロック方式等がある。

【０００５】従来の排他制御方法では、ロック待ち方式は、計算機システム全体あるいはリソース毎に固定的に定められており、オペレーティングシステムは、固定的に定められているロック待ち方式の処理を行う。

【０００６】

【発明が解決しようとする課題】一般的に、共有メモリ方式の多重プロセッサ計算機システムでは、最も単純なスピン方式を採用することが多いが、本質的にどの方式を採用すべきかは、その排他対象となるリソースの性質に依存している。

【０００７】ここで、ロック待ちが発生した場合に、計算機システムの性能を落とさないために留意すべき点は、ロックワードに対して必要以上に頻繁にアクセスを行わないようにすることである。すなわち、ロックワードに対して必要以上に頻繁にアクセスを行うと、メモリバンク競合による性能低下を招いてしまうので、そのようなことは避けなければならない。従って、ロック待ちが発生した場合の最も効率的な対処方法は、ロックを取っているプロセスがリソースをアンロックしたタイミングで、タイムリにロックを取得できるような待ち方で一定時間待つ方式である。この待ち時間は、具体的にはそのリソースに対するアクセスパターン、即ちロックとアンロックとのタイミング、さらに言えばロックされていた時間（ロック粒度）に依存していると考えられる。

【０００８】ロック粒度が十分に小さい場合は、（１）のスピン方式が適している。また、ロック粒度がスリープ処理に要する時間（即ちスイッチ処理時間）よりも十分に大きい場合で、なおかつ他に行うべき処理がない場合は、（２）のスリープ方式が適している。逆に、他に行うべき処理がある場合は、（３）のトライロック方式が適しているということになる。つまり、ロック粒度が十分に小さい場合は、ＣＰＵ時間は無駄に消費されるが、その時間は極僅かなので、制御が簡単で負荷の少ない（１）のスピン方式が適している。また、ロック粒度がスイッチ処理時間よりも十分に大きい場合は、制御は複雑になるが、ＣＰＵ時間を有効に使用することができるので、（２）のスリープ方式が適している。

【０００９】しかしながら、ロック粒度は、ロック毎に一定の値になるわけではなく、システムの運用状況に応じて動的に変動する。このため、計算機システム全体あるいはリソース毎に固定的にロック待ち方式が定められている従来の排他制御方法では、ロック粒度に応じた最適なロック待ち方式の処理が行われず、その結果、計算機システムの性能を低下させてしまう場合があるという問題がある。例えば、スピン方式による処理が固定的に割り当てられているリソースのロック粒度が、システム運用に伴って大きくなった場合には、ロックワードに対するアクセスが必要以上に頻繁に行われ、メモリバンク競合による性能低下を招いてしまう。また、例えば、スリープ方式による処理が固定的に割り当てられているリソースのロック粒度がシステム運用に伴ってスイッチ処理時間よりも小さくなった場合には、スピン方式に比較して制御が複雑になるだけで、ＣＰＵ時間を有効利用できなくなってしまう。

【００１０】そこで、本発明の目的は、ロック粒度の統計情報を利用してロック待ち方式を動的に変更することで、メモリバンク競合や、無駄なスイッチ処理等によるオーバヘッド時間を減少させることにより、システム全体の性能を向上させることにある。

【００１１】尚、特開平４－２９７９４９号公報には、競合の発生回数等の統計情報に基づいて、排他単位のサイズを変更する技術が記載されているが、ロック粒度の統計情報に基づいてロック待ち方式を動的に変更するものではない。

【００１２】

【課題を解決するための手段】本発明のロック粒度統計情報を利用した排他制御方法は、上記目的を達成するため、複数のプロセッサがメインメモリを共有する共有メモリ方式の多重プロセッサ計算機システムにおいて、前記複数のプロセッサによって共有されるリソースがロックされる毎にロック粒度を算出すると共に、該算出したロック粒度に基づいて前記リソースのロック粒度統計情報を更新し、前記リソースに対するロック待ちが発生したとき、前記リソースのロック粒度統計情報に基づいてロック待ち方式を決定する。

【００１３】ここで、ロック粒度統計情報には、例えば、ロック粒度の平均値が含まれ、また、ロック粒度統計情報に基づいて決定するロック方式には、例えば、スピン待ち時間経過後に再度リソースに対するロックを試みるスピン方式或いはロック要求元をスリープさせるスリープ方式が含まれる。

【００１４】また、本発明のロック粒度統計情報を利用した排他制御方法は、ロック方式としてスピン方式が決定された場合、スピン待ち時間を最適なものにできるようにするため、ロック待ち方式をスピン方式と決定した場合は、前記リソースのロック粒度統計情報に基づいて前記ロック待ち時間を決定する。

【００１５】

【発明の実施の形態】次に本発明の実施の形態について図面を参照して詳細に説明する。

【００１６】図１は、本発明のロック粒度統計情報を利用した排他制御方法が適用される共有メモリ方式の多重プロセッサ計算機システムの構成例を示すブロック図であり、複数台のプロセッサ（コンピュータ）１－１～１－ｎと、各プロセッサ１－１～１－ｎによって共用され

るメインメモリ２と、複数台の入出力処理装置３－１～３－ｍとから構成される。

【００１７】メインメモリ２上には、処理の単位となるプロセス４－１～４－ｉと、オペレーティングシステム５とが存在する。オペレーティングシステム５は、排他単位となるリソース（メモリ資源やＩ／Ｏ資源）毎の管理テーブル６－１～６－ｊによって管理される各リソース毎のロックワード７－１～７－ｊ及び待ち行列８－１～８－ｊとを含んでいる。

【００１８】図２，図３はオペレーティングシステム５の処理例を示す流れ図であり、以下各図を参照して本実施例の動作を説明する。

【００１９】今、例えば、プロセッサ１－１上で動作しているプロセス４－１において、管理テーブル６－ｊで管理されているリソースＲｊ（図示せず）に対する排他的なアクセスが必要になった場合、プロセス４－１は、リソースＲｊのロックを要求するシステムコールを発行する。尚、このシステムコールには、プロセス４－１において、リソースＲｊを使用しない他に行うべき処理があるか否かを示すフラグ情報が含まれる。

【００２０】上記したシステムコールが発行されると、オペレーティングシステム５は、何れかのプロセッサ（例えばプロセッサ１－ｎ）上で、図２の流れ図に示す処理を行う。つまり、オペレーティングシステム５は、プロセッサ１－ｎによって読み取られ、プロセッサ１－ｎの動作を制御することで、図２の流れ図に示す処理を行わせる。

【００２１】先ず、ステップＳ１のロック判定処理において、オペレーティングシステム５は、テストアンドセット命令を実行し、リソースＲｊに対するロックワード７－ｊを参照すると共にロックワード７－ｊにロック状態であることを示す値を書き込む。そして、参照時のロックワード７－ｊの値がアンロック状態であることを示している場合、即ちリソースＲｊがロック可能である場合は、ステップＳ２のロック処理を行い、参照時のロックワード７－ｊの値がロック状態であることを示している場合、即ちリソースＲｊがロック不可の場合は、ステップＳ３のロック待ち方式判定処理を行う。

【００２２】今、例えば、ステップＳ１において、ロック可能と判定した場合は、オペレーティングシステム５は、ステップＳ２のロック処理において、ロック要求元のプロセス４－１に対してロック成功を通知し、更に現在時刻を取得しそれをロック開始時刻としてリソースＲｊ対応の管理テーブル６－ｊに書き込む。

【００２３】ロック要求元のプロセス４－１は、ロック成功が通知されると、リソースＲｊを独占的に使用した処理を開始する。

【００２４】その後、プロセス４－１が、リソースＲｊのアンロックを要求するシステムコールを発行すると、オペレーティングシステム５は何れかのプロセッサ（例えばプロセッサ１－ｎ）上で図３の流れ図に示す処理を行う。つまり、オペレーティングシステム５はプロセッサ１－ｎによって読み取られ、プロセッサ１－ｎの動作を制御することで、図３の流れ図に示す処理を行わせる。

【００２５】先ず、ステップＳ１１のアンロック処理において、オペレーティングシステム５は、アンロック要求されたリソースＲｊに対応するロックワード７－ｊにアンロック状態を示す値を書き込む。

【００２６】次いで、ステップＳ１２のロック粒度算出処理において、リソースＲｊ対応の管理テーブル６－ｊに格納されているロック開始時刻と現在時刻との差分を求めることにより、リソースＲｊがロックされていた時間（ロック粒度）を算出する。

【００２７】その後、ステップＳ１３のロック粒度統計情報算出処理において、管理テーブル６－ｊに格納されている、リソースＲｊについてのロック粒度統計情報を更新する。ロック粒度統計情報には、ロック回数，ロック粒度の平均値，分散値，最大値，最小値等が含まれており、ステップＳ１３においては、ロック回数を＋１したり、ステップＳ１２で算出した今回のロック粒度と管理テーブル６－ｊに登録されているロック粒度統計情報とに基づいてロック粒度の平均値，分散値，最大値，最小値等を更新する。

【００２８】そして、最後にステップＳ１４のウェイクアップ処理において、リソースＲｊのロック待ちをしている、スリープしているプロセスの識別子が登録される待ち行列８－ｊを参照し、識別子が登録されている場合は、例えば、先頭に登録されている識別子を取り出し、その識別子のプロセスを起こす。つまり、プロセスをレディ状態にし、ＣＰＵ時間の割り当て対象にする。

【００２９】次に、図２のステップＳ１において、プロセス４－１によってロック要求されたリソースＲｊをロックできないと判定した場合の動作について説明する。

【００３０】ステップＳ１において、リソースＲｊをロックできないと判定した場合、オペレーティングシステム５は、ステップＳ３のロック待ち方式判定処理を行い、ロック待ち方式を決定する。

【００３１】このステップＳ３のロック待ち方式判定処理では、例えば、以下のようにしてロック待ち方式を決定する。

【００３２】先ず、プロセス４－１が発行したシステムコールに含まれているフラグ情報を参照し、プロセス４－１に、リソースＲｊを使用しない他に行うべき処理があるか否かを調べる。そして、他に行うべき処理がある場合は、ロック待ち方式をトライロック方式とする。これに対して、他に行うべき処理がない場合は、リソースＲｊ対応の管理テーブル６－ｊに格納されているリソースＲｊのロック粒度統計情報に基づいて、ロック待ち方

式をスピン方式にするか、スリープ方式にするかを決定する。例えば、ロック粒度の平均値がスリープ処理に要する時間よりも短い場合はスピン方式、長い場合はスリープ方式、等しい場合はスピン方式、スリープ方式の内の予め定められている一方の方式とする。更に、ロック待ち方式をスピン方式と決定した場合には、管理テーブル６－ｊに登録されているリソースＲｊのロック粒度統計情報に基づいて、スピン待ち時間を決定する。例えば、ロック粒度の平均値或いは最小値をスピン待ち間隔とする。

【００３３】ステップＳ３でロック待ち方式をスピン方式にすることを決定した場合は、オペレーティングシステム５は、ステップＳ４のスピン待ち処理を行う。ステップＳ４のスピン待ち処理では、ステップＳ３で決定されたスピン待ち時間が経過するのを待ち、スピン待ち時間が経過すると再びステップＳ１のロック判定処理を行う。

【００３４】また、ステップＳ３でロック待ち方式をスリープ方式にすることを決定した場合は、ステップＳ５のスリープ待ち処理を行う。このステップＳ５のスリープ待ち処理では、ロック要求元のプロセス４－１をスリープさせ、プロセス４－１が割り当てられていたプロセッサ１－１を開放する。即ち、プロセス４－１をウエイト状態にすることにより、それが割り当てられていたプロセッサ１－１を開放する。更に、ステップＳ３では、プロセス４－１の識別子を、プロセス４－１がロック要求したリソースＲｊに対応する待ち行列８－ｊに登録する。

【００３５】また、ステップＳ３でロック待ち方式をトライロック方式にすることを決定した場合は、ステップＳ６のトライロック処理を行う。このステップＳ６のトライロック処理では、ロック要求元のプロセス４－１にロック失敗を通知する。この通知を受けたプロセス４－１は、リソースＲｊを使用しない他の処理を実行する。

【００３６】図４は図３のロック粒度統計情報算出処理（ステップＳ１３）および図２のロック待ち方式判定処理（ステップＳ３）の処理を詳細に説明するための図である。

【００３７】プロセスＮがリソースＲ（図示せず）をアンロックすると、ステップＳ１２のロック粒度算出処理において、ロック粒度Ｔｎを算出し、次のＳ１３のロック粒度統計情報算出処理において、前回までのリソースＲのロック粒度統計情報Ｐｎ-１と今回新たに算出したロック粒度Ｔｎとから新たなロック粒度統計情報Ｐｎを次式（１）により算出する。

【００３８】Ｐｎ＝関数Ｆ１(Ｐｎ-１,Ｔｎ)　…　（１）

【００３９】ここでロック粒度統計情報Ｐには、tmin＝ロック粒度の最小値，tmax＝ロック粒度の最大値，ｔ＝ロック粒度の総和，tc＝ロック回数，t2＝ロック粒度の２乗の総和，tav＝ロック粒度の平均値，tbu＝ロック粒度の分散値，その他の情報が含まれている。すなわち、Ｐ＝（tmin，tmax，t，tc，t2，tav，tbu，その他）である。

【００４０】プロセスＮがリソースＲをロックしているときに、プロセスＮ＋１がリソースＲをロックしようとすると、図２のステップＳ１のロック判定処理においてロック不可と判定され、ステップＳ３のロック待ち方式判定処理が行われる。このステップＳ３のロック待ち方式判定処理では、他に行うべき処理があるかないかを示すフラグ情報flagとロック粒度統計情報Ｐｎ とから次式（２）に示すようにしてロック待ち方式Ｍｎ+１ を決定する。

【００４１】Ｍｎ+１ ＝関数Ｆ２(Ｐｎ，flag)　…　（２）

【００４２】ロック待ち方式をスピン方式と決定した場合には、ロック粒度統計情報Ｐｎ から次式（３）に示すようにしてスピン待ち時間Ｗｎ+１ を決定する。

【００４３】Ｗｎ+１ ＝関数Ｆ３(Ｐｎ)　…　（３）

【００４４】

【発明の効果】以上説明したように、本発明は、ロック粒度統計情報を利用してロック待ち方式を動的に変更するので、メモリバンク競合や、無駄なスイッチ処理等によるオーバヘッド時間を減少させることができる。この結果、共有メモリ方式の多重プロセッサ計算機システムの性能を向上させることができる。

【００４５】更に、本発明は、ロック待ち方式をスピン方式と決定した場合には、ロック粒度統計情報を利用してスピン待ち時間を決定するようにしているので、メモリバンク競合によるオーバヘッド時間を更に減少させることができる。

【図面の簡単な説明】

【図１】本発明方法を適用する共有メモリ方式の多重プロセッサ計算機システムの構成例を示すブロック図である。

【図２】オペレーティングシステム５のロック要求時の処理例を示す流れ図である。

【図３】オペレーティングシステム５のアンロック要求時の処理例を示す流れ図である。

【図４】ロック待ち方式判定処理およびロック粒度統計情報算出処理を説明するための図である。

【符号の説明】

１－１～１－ｎ…プロセッサ

２…メインメモリ

３－１～３－ｍ…入出力処理装置

４－１～４－ｉ…プロセス

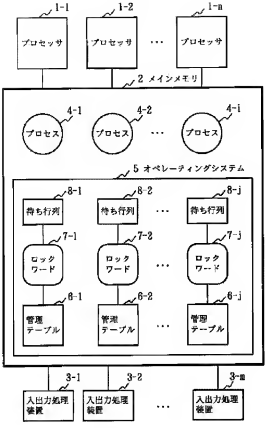
５…オペレーティングシステム

６－１～６－ｊ…管理テーブル

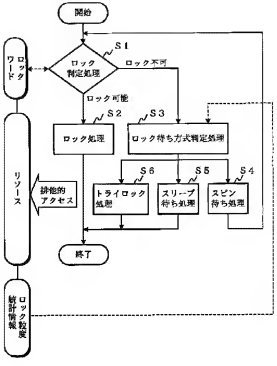
７－１～７－ｊ…ロックワード

８－１～８－ｊ…待ち行列

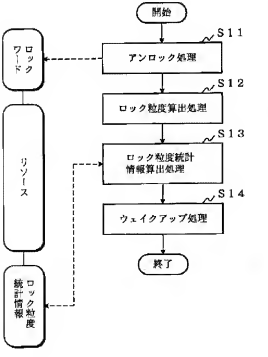
【図1】


1-1
プロセッサ
1-2
プロセッサ
1-n
プロセッサ
2 メインメモリ
4-1
プロセス
4-2
プロセス
4-i
プロセス
5 オペレーティングシステム
8-1
待ち行列
8-2
待ち行列
8-j
待ち行列
7-1
ロックワード
7-2
ロックワード
7-j
ロックワード
6-1
管理テーブル
6-2
管理テーブル
6-j
管理テーブル
3-1
入出力処理装置
3-2
入出力処理装置
3-m
入出力処理装置

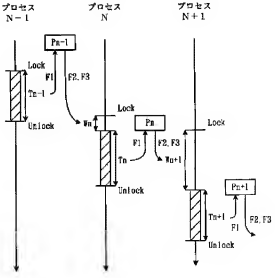
【図2】


開始
ロックワード
S1
ロック判定処理
ロック不可
ロック可能
S2
ロック処理
S3
ロック待ち方式判定処理
リソース
S6
S5
S4
排他的アクセス
トライロック処理
スリープ待ち処理
スピン待ち処理
終了
ロック粒度統計情報

【図3】


開始
ロックワード
S11
アンロック処理
S12
ロック粒度算出処理
S13
ロック粒度統計情報算出処理
S14
ウェイクアップ処理
終了
リソース
ロック粒度統計情報

【図4】


プロセス N−1
プロセス N
プロセス N+1
Ps-1
Lock
F1
F2, F3
Tn-1
Unlock
Wn
Lock
Ps
F1
F2, F3
Tn
Wn+1
Unlock
Lock
Ps+1
Tn+1
F2, F3
F1
Unlock